The FIRST VIRTUAL (TM) Internet Payment System

Technical Overview

Outline

Basic Principles for Internet Information Commerce
- Openness
- Risk Management
- Authentication
- Privacy

System Architecture
- Data structures
- Client Interfaces

Connection Protocols: Mail and SGCP

The application/green-commerce MIME Type
- The data fields
- The transaction types

Additional Clever Hacks

Basic Principles for Internet Information Commerce

Openness

Must create open playing field for all buyers & sellers

Must create equal opportunities to all writers of support software

Bottom line: must permit fair competition wherever feasible

Note: Traditional credit cards are a *closed* system.

"As open as cash"

Basic Principles for Infocommerce, con’t

Risk Management

Nothing is risk-free.

Risks must be limited and well-understood

Buyers’ risks must be considered as well as sellers’ risks

Where feasible, supplemental risk management strategies should be optional.

"As safe as credit cards"

Basic Principles for Infocommerce, con’t

Authentication

Authentication infrastructure must be universal.

Implication: encryption is problematic. (Worst case: consider France.)

Authentication does NOT imply digital signatures.

"As authenticated as credit cards"

Basic Principles for Infocommerce, con't

Privacy

Privacy must be protected.

Different levels of concern for privacy must be respected.

Abstract concerns for privacy probably won't make me tolerate a complex interface for buying gum.

Ease of use probably won't make me buy AIDS information in public.

Implication: Privacy can come in layers.

Lowest layer: Privacy through obscurity

Highest layer: Public key encryption.

"As private as you want to be"

The First Virtual System Architecture

Client/server protocols based on TCP/IP standards

The Basic Model

Client sends a "transaction" to FV server.

(A transaction is NOT necessarily a fund transfer)

FV server gives response to client

FV server may take additional actions.

Data Structures

All transactions/responses are represented as structured MIME data.

For some purposes, standard multipart/mixed, multipart/alternative.

One new specialized MIME type.

Client Interfaces

All access may be done via email (SMTP)

Optional access through interactive protocol (SGCP)

Specialized uses of finger, telnet, DNS, etc.

Connection Protocols: Mail and SGCP

Mail is simplest:

To: sgcs@card.com
Content-type: application/green-commerce;
transaction=transfer-request

SGCP is almost as easy

New protocol (port 440), modeled on POP protocol

Specification published openly
-- on ftp.fv.com today
-- soon-to-be an Informational RFC

Basic function: interactive exchange of MIME objects.

Client passes one MIME object to server.

Server gives back "+ERR" or "+OK"

Sometimes, server also gives back another MIME object.

The application/green-commerce MIME Type

A new MIME type for information commerce.

Specification published openly
- -- on ftp.fv.com today
- -- soon-to-be an Informational RFC

One mandatory Content-type parameter:
- -- "Transaction" tells what kind of action

Simple data structure:
- -- Modelled on RFC 822
- -- "Attribute: Value" syntax
- -- RFC 822 parsers are instantly reusable

A Simple Example

```
To: sgcs@card.com
Content-type: application/green-commerce;
    transaction=transfer-request

BUYER: Joe Is a cool dude
SELLER: CrazyRDIM
AMOUNT: 19.99
CURRENCY: USD US Dollars
TRANSFER-ID: <CR42@somewhere.com>
SECURITY-REQUIREMENTS: None
DESCRIPTION: Purchase of used stock quotes

We gave you lots of useful stock quotes and
tips that may have helped you in your
investments, and it isn’t our fault if a
grand jury is interested in you!
```

The data fields in application/green-commerce data

Data fields, or "attributes", are like header fields in RFC 822.

Typical attributes:

ACCOUNT-ID
AMOUNT
BUYER
BUYER-NAME
CURRENCY
EMAIL-ADDRESS
ERROR-CODE
EXPLANATION
FINANCIAL-CHANGES
FULL-NAME
PREFERRED-CURRENCY
PREFERRED-LANGUAGE
SELLER
SELLER-NAME
SERVER-ID
SERVICE-CHARGE

The application/green-commerce Transaction Types

The "transaction" parameter describes the type of action -- a SINGLE STEP in a protocol exchange.

There are 26 such transactions.

It usually takes more than one to perform what a user would think of as a "transaction".

Here, we explain them in groups of related transactions.

Account Setup Transactions

1. User requests an account application:
   APPLICATION-REQUEST
   Key Attributes: none

2. Server sends back a fill-in form:
   APPLICATION-RESULT
   Key Attributes: none

3. User fills in form and sends it back:
   NEWACCT-REQUEST
   Key Attributes: many
   Form from #2 is structured to help create this.

4. Server sends back result:
   NEWACCT-RESULT
   Key Attribute: APPLICATION-STATUS
   May send more than one (pending, accepted)

Sensitive financial information entered through ARU (Audio Response Unit)

Fund transfer (purchase)

1. Some user or application requests a transfer:
   TRANSFER-REQUEST
   Key Attributes: BUYER, SELLER, AMOUNT, CURRENCY, DESCRIPTION

2. Server sends buyer email seeking confirmation:
   TRANSFER-QUERY
   Key Attributes: BUYER-NAME, SELLER-NAME, AMOUNT, CURRENCY, DESCRIPTION, SERVER-ID

3. User responds yes/no/fraud:
   TRANSFER-RESPONSE
   Key Attribute: AUTHORIZATION, SERVER-ID
   Special hack: SERVER-ID in Subject.

4. Server tells seller/initiator of authorization status:
   TRANSFER-RESULT
   Key Attributes: From #2, #3.

Account Status Inquiries

1. Customer asks server about account status
   INQUIRY-REQUEST
   Key Attribute: ACCOUNT-ID

2. Server responds active/invalid/suspended/seller-only:
   INQUIRY-RESULT
   Key Attributes: ACCOUNT-ID, ACCOUNT-STATUS

Server Capability Inquiries

1. Customer asks server about server capabilities
   CAPABILITIES-REQUEST
   Key Attributes: None.

2. Server responds with list of supported features:
   CAPABILITIES-RESULT
   Key Attributes: CURRENCY-TYPES, LANGUAGES, TRANSACTION-TYPES, TRANSFER-TYPES

Tracking the Flow of Money

At each stage of processing, the relevant parties are notified of the status of their financial transactions with various application/green-commerce transaction types:

Billing your credit card: PAYIN-NOTIFICATION

Crediting your bank account: PAYOUT-NOTIFICATION

Receiving a chargeback from the buyer:
To buyers: PAYIN-CHARGEBACK-NOTIFICATION
To sellers: PAYOUT-CHARGEBACK-NOTIFICATION

Failure to settle (e.g. credit card no longer good):
To buyers: PAYIN-FAILURE-NOTIFICATION
To sellers: COLLECTION-FAILURE-NOTIFICATION

Changing Account Information

Information about an account is changed using six transactions that are modeled on account setup & fund transfer:

INITCHG-REQUEST
INITCHG-RESULT
CHGACCT-REQUEST
CHGACCT-QUERY
CHGACCT-RESPONSE
CHGACCT-RESULT

Anything but sensitive financial information can be changed on the net.

Changes to email address require confirmation through the old address.

Changes to financial information require another call to the ARU.

Miscellany

HISTORY-REQUEST

Key parameter: ACCOUNT-ID

Generates a history of all account activity.

Sends a MIME multipart to the email address of record.

ERROR-REPORT

Provides details of any transaction the server could not process.

Additional Clever Hacks

For testing: test.card.com

For mail to accountholders:
accountid@relay.card.com
(Normalize account id to alphanumerics)

For quick inquiry-request:
finger account-id@card.com

For brain-damaged gateways:

localpart%domain@lotus-notes.challenged.card.com

For Additional Information

URL http://www.fv.com
By email: info@fv.com

Full Specs available:
ftp.fv.com: pub/docs/*-spec.{ps,txt}

FREE code available for:
ftp server mods to sell by ftp
CGI scripts to sell by WWW
Shell & C interfaces for basic transactions
Mail reader extensions for nicer user interface
ftp.fv.com: pub/code/*/*

There’s no catch -- it is REAL TODAY!